Chapitre 1 : Complexité des problèmes de décision

ENSIIE - Théorie de la complexité

Dimitri Watel (dimitri.watel@ensiie.fr)

2022

Problème de décision

Définition (informelle)

Un problème de décision est un problème possédant des entrées (*les instances*) et une question concernant l'entrée dont la réponse est **Oui** ou **Non** en fonction de l'entrée.

P, NP et NP-Complet en bref

- P : problèmes de décision qu'on peut résoudre en temps polynomial ;
- NP : problèmes de décision où quand on nous montre une preuve que la réponse est OUI, on peut la *vérifier* en temps polynomial ;
- NP-Complet : problèmes de décision de NP les plus difficiles..

En particulier : si un problème NP-Complet est dans P, alors

$$\mathrm{P} = \mathrm{NP}.$$

C'est peu probable : démontrer qu'un problème est NP-Complet montre qu'il est inutile de s'évertuer à démontrer qu'il est dans P.

Complexité d'un algorithme

Définition

- La complexité en temps d'un algorithme est le nombre d'opérations élémentaires nécessaires avant qu'il ne se termine.
- La complexité en espace d'un algorithme est le nombre de cases mémoires utilisées avant qu'il ne se termine.

Soit un algorithme $\mathcal{A}$ et $\mathcal{I}$ une entrée de $\mathcal{A}$, on pose $t(\mathcal{A}, \mathcal{I})$ la complexité en temps de $\mathcal{A}$ quand l'entrée est $\mathcal{I}$. De même, on pose $s(\mathcal{A}, \mathcal{I})$ sa complexité en espace.

La complexité est souvent calculée en fonction de la taille de l'entrée.

Taille de l'entrée usuelle

Définition

La taille de l'entrée est définie comme le nombre de bits nécessaires pour l'encoder.

Exemples

- pour un graphe : nombre de nœuds, nombre d'arêtes
- pour une liste : nombre d'éléments
- pour un entier k : $\log_2(k)$
- pour une liste L d'entiers : $\sum_{k \in L} \log_2(k)$
- ...

On note généralement $|x|$ ou n cette taille.

Complexité dans le pire cas

Définition

Soit un algorithme $\mathcal{A}$ et $\mathcal{I}_n$ toutes les entrées de $\mathcal{A}$ de taille au plus n, la complexité (en temps) dans le pire cas de $\mathcal{A}$ est une fonction f telle que

$$f(n) = \max_{\mathcal{I} \in \mathcal{I}_n}(t(\mathcal{A}, \mathcal{I}))$$

On définit de même la complexité en espace dans le pire cas.

Autres complexités

- Complexité en moyenne
- Complexité amortie
- Complexité lisse

Complexité asymptotique dans le pire cas

Définition

Soit un algorithme $\mathcal{A}$ dont la complexité dans le pire cas est f, on dit que $\mathcal{A}$

- a une complexité dans le pire cas bornée asymptotiquement par g si $f(n) = O(g(n))$ quand $n \to +\infty$.
- a une complexité dans le pire cas minorée asymptotiquement par g si $f(n) = \Omega(g(n))$ quand $n \to +\infty$.
- a une complexité dans le pire cas asymptotiquement de l'ordre de g si $f(n) = \Theta(g(n))$ quand $n \to +\infty$.

Par abus de langage, on dira que $\mathcal{A}$ a une complexité $O(g(n))$ si la **complexité en temps dans le pire cas** de $\mathcal{A}$ est bornée asymptotiquement par g.

Complexités usuelles

- Polynomiale : $O(n)$, $O(n^2)$, $O(n^3)$, $O(n^c)$
- Exponentielle : $O(2^n)$, $O(3^n)$, $O(n!)$, $O(2^{n^2})$, $O(2^{(n^c)})$
- Constante : $O(1)$
- Linéaire : $O(n)$
- Quadratique : $O(n^2)$
- Cubique : $O(n^3)$
- Logarithmique : $O(\log(n))$
- Polylogarithmique : $O(\log^c(n))$
- Sous-exponentielle : $O(2^{\log^c(n)})$
- Superexponentielle : $O(2^{(2^n)})$

Problème de décision

Définition (formelle)

Un problème de décision Π est constitué d'un ensemble $\mathcal{L}$, dont les éléments sont appelés *instances* ou *entrées*, et d'un ensemble $\mathcal{L}_Y \subset \mathcal{L}$ d'*instances positives*. On nomme $\mathcal{L} \backslash \mathcal{L}_Y = \mathcal{L}_N$ les *instances négatives*.

Un algorithme résolvant Π est capable, pour toute instance $\mathcal{I} \in \mathcal{L}$ de décider si $\mathcal{I} \in \mathcal{L}_Y$ ou si $\mathcal{I} \in \mathcal{L}_N$ en temps fini.

Complexité d'un problème de décision

Définition

Soit un problème de décision Π, alors Π a une complexité (au pire) $O(f(n))$ s'il existe un algorithme $\mathcal{A}$ de complexité $O(f(n))$ pour le résoudre.

Définition de la classe P

Un problème de décision Π est dit polynomial ou appartenant à la classe P si sa complexité est polynomiale. Autrement dit, il existe une constante c telle que la complexité de Π est $O(n^c)$.

La classe de complexité EXPTIME

Définition de la classe EXPTIME

Un problème de décision Π est dit exponentiel ou appartenant à la classe EXPTIME si sa complexité est exponentielle. Autrement dit, il existe une constante c telle que la complexité de Π est $O(2^{(n^c)})$.

$$\text{P} \subsetneq \text{EXPTIME}$$

La classe de complexité NP

Définition (informelle) de la classe NP

Un problème de décision Π est appartient à la classe NP si, quand la réponse est **OUI**, il existe une preuve vérifiable en temps polynomial.

! ! !

$$P \subset NP \subsetneq EXPTIME$$

Vérifier une solution

Définition d'un vérifieur

Soit un problème de décision Π, un vérifieur $\mathcal{V}$ pour les réponses **OUI** de Π de complexité f et g est un **algorithme**

- qui prend en entrée
  - une instance $\mathcal{I}$ de taille n de Π
  - un *certificat* w (une suite de 1 et de 0) de taille $O(g(n))$
- qui répond **OUI** ou **NON** en temps $O(f(n))$
- qui répond **OUI** pour au moins un certificat w si la réponse à $\mathcal{I}$ est **OUI**
- qui répond **NON** pour tout certificat w si la réponse à $\mathcal{I}$ est **NON**

Définition de la classe NP

Un problème de décision Π est appartient à la classe NP s'il existe une constante c et s'il existe un vérifieur $\mathcal{V}$ pour les réponses **OUI** de complexité $O(n^c)$ et $O(n^c)$.

! ! !

$$\mathrm{P} \subset \mathrm{NP} \subsetneq \mathrm{EXPTIME}$$